

Kotlin - Environment Setup

Installing Kotlin command-line compiler

One of the key features of Kotlin is that it has interoperability with Java i.e. You can write Kotlin and Java code in the same application. Like Java, Kotlin also runs on JVM therefore to install Kotlin on Windows directly and work with it using the command line You need to make sure you have JDK installed in your system.

Verifying the Java installation

To verify Java installation –

- Open command prompt and verify the current version of Java using the **javac version** command –

```
C:\Users\TP>javac -version
javac 1.8.0_261
```

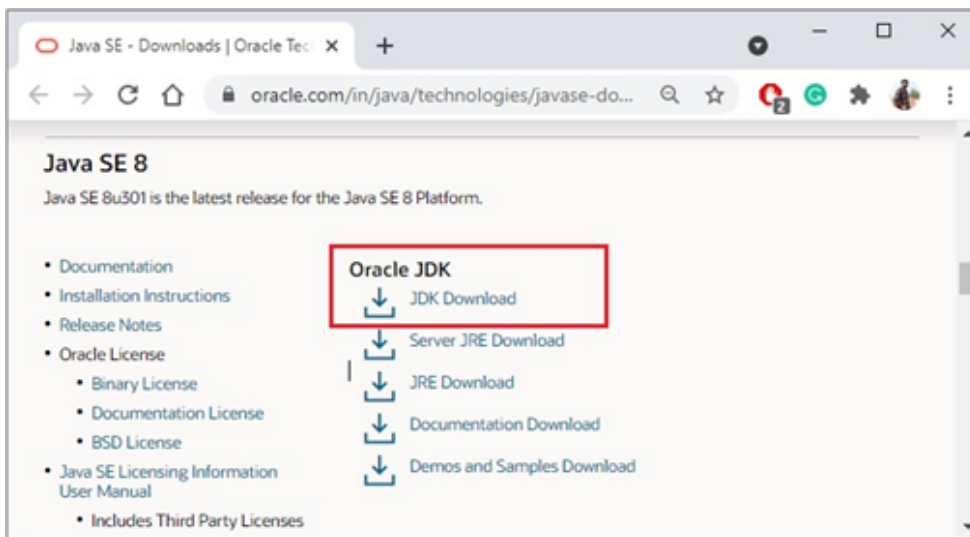
If you doesn't have Java installed in your system it generates the following error

```
C:\Users\Krishna Kasyap>javac -v
'javac' is not recognized as an internal or external command,
operable program or batch file.
```

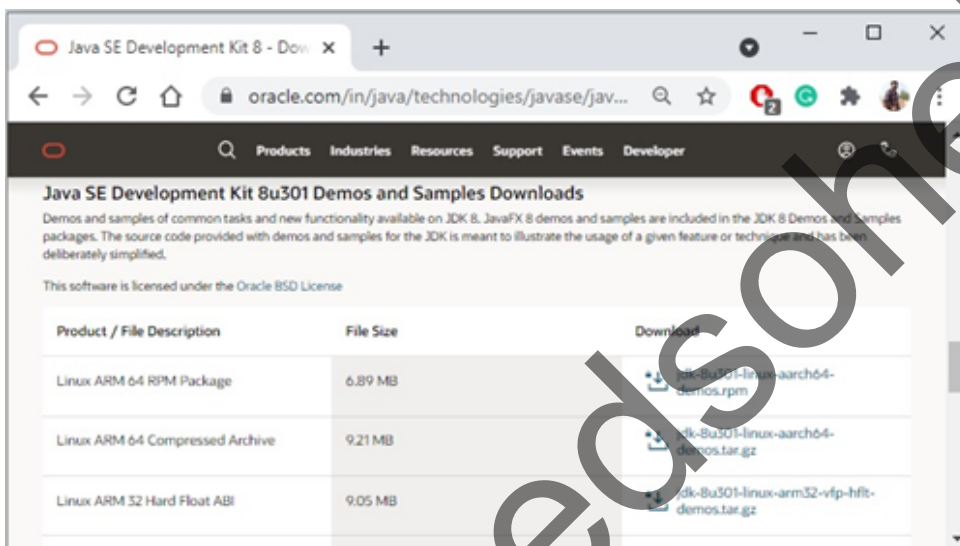
You can install JDK by following the steps given below

Installing JDK8

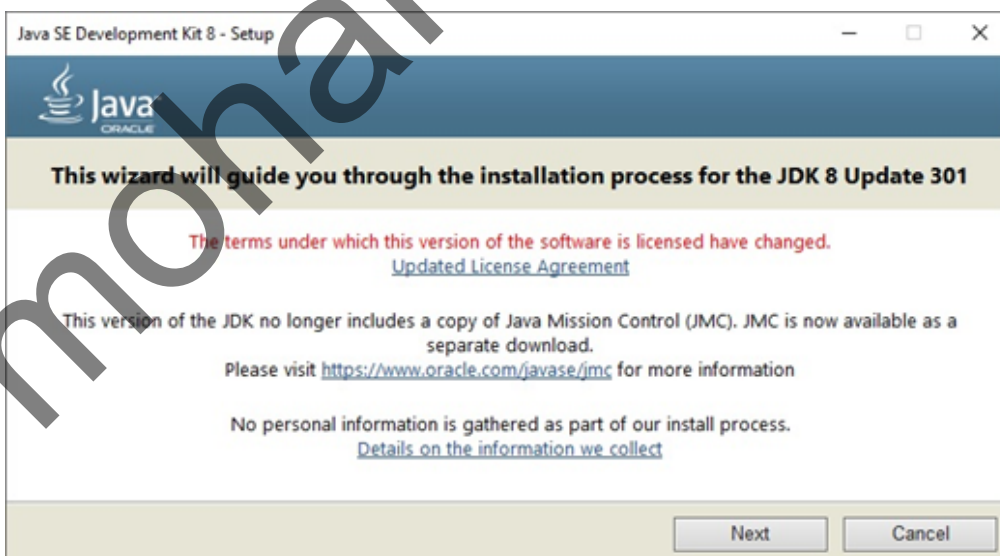
- Open the following Oracle Java Downloads page.
- Click on the **JDK Download** link under **Java SE 8** section.



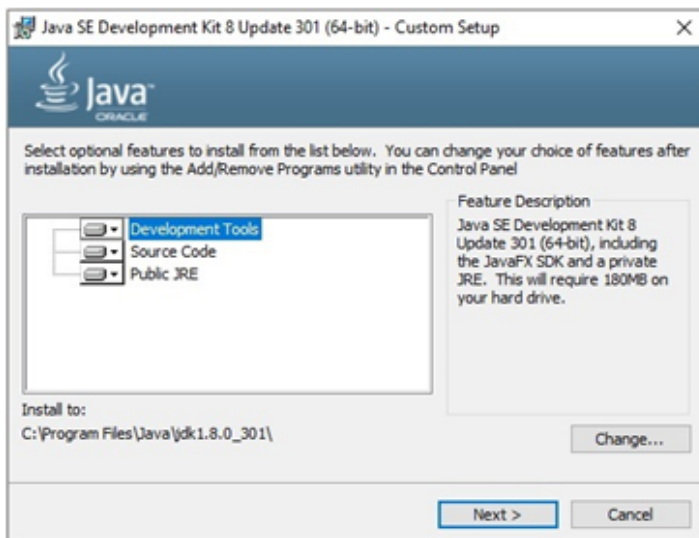
- This will redirect to the page that contains JDK software for various platforms, select the desired version (.exe) and download it.



- After downloading the file JDK file (assume we have downloaded `jdk_windows-x64_bin.exe`), start the installation by running it.



- By default, Java will be installed in the path `C:\Program Files\Java\jdk1.8.0_301\` you can change the path by clicking on the Change... button.



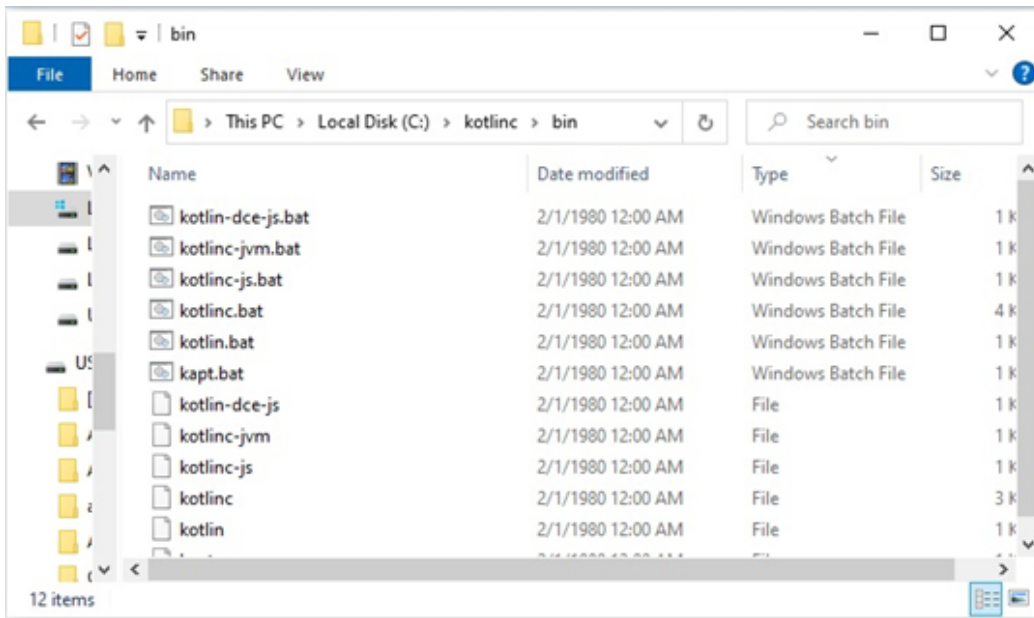
- After the completion of the installation click on the **Close** button.



Kotlin Command line compiler

Kotlin command line compiler is available at the JetBrains Kotlin GitHub releases page .

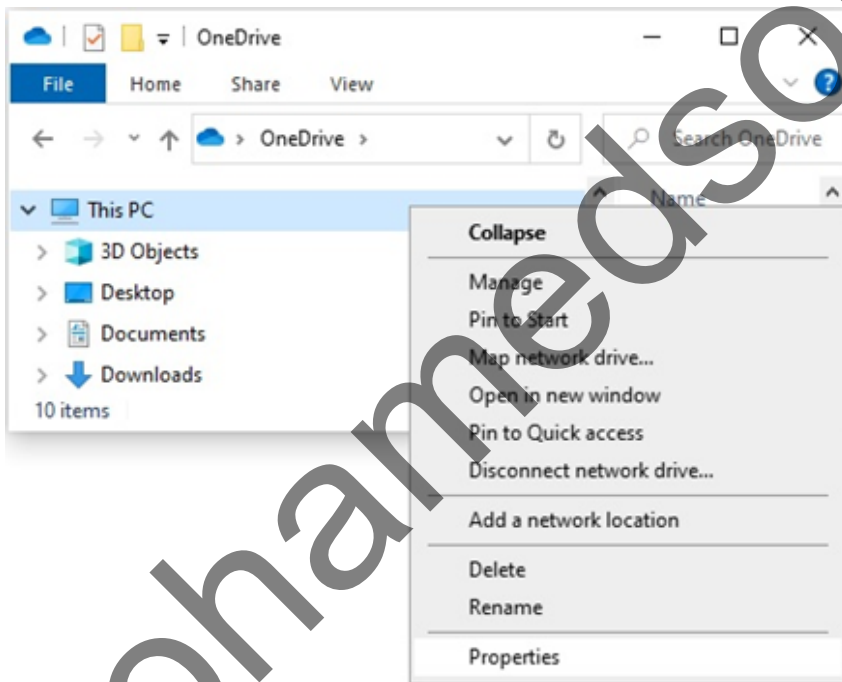
- Download the latest version.
- Unzip the downloaded file and place it in the desired folder.
- The **Bin** directory of the downloaded folder contains all the binary files to run Kotlin.



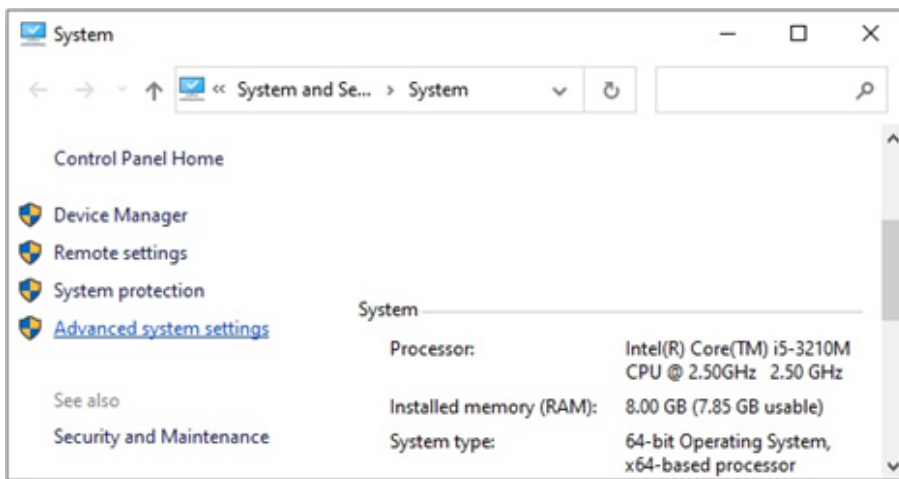
- Now, set Path environment variable to this folder.

Setting the Path variable

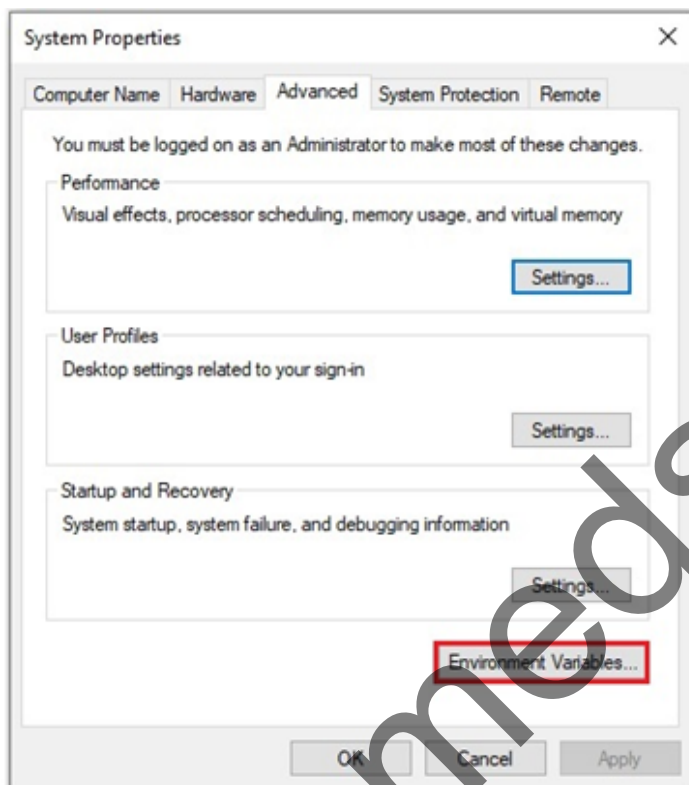
- Right click on My computer or This PC, select **Properties**.



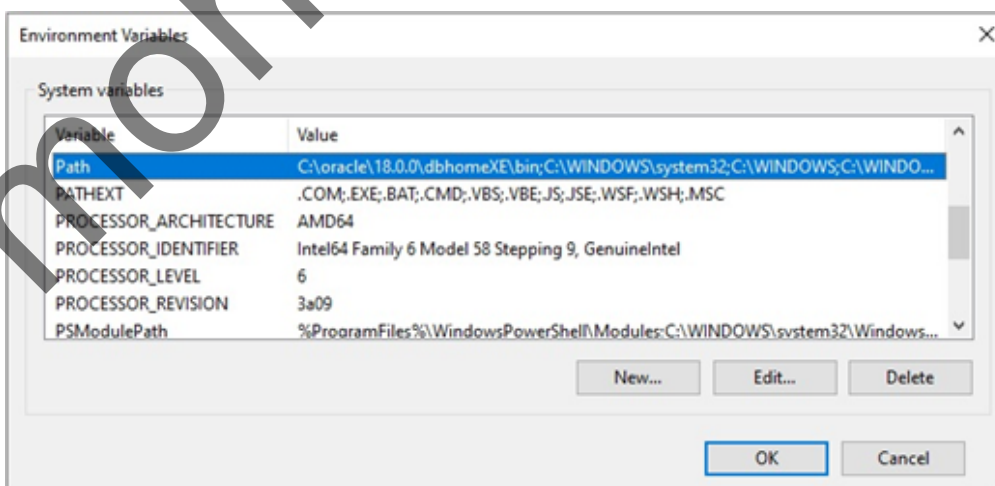
- Click on **Advanced System Settings**.



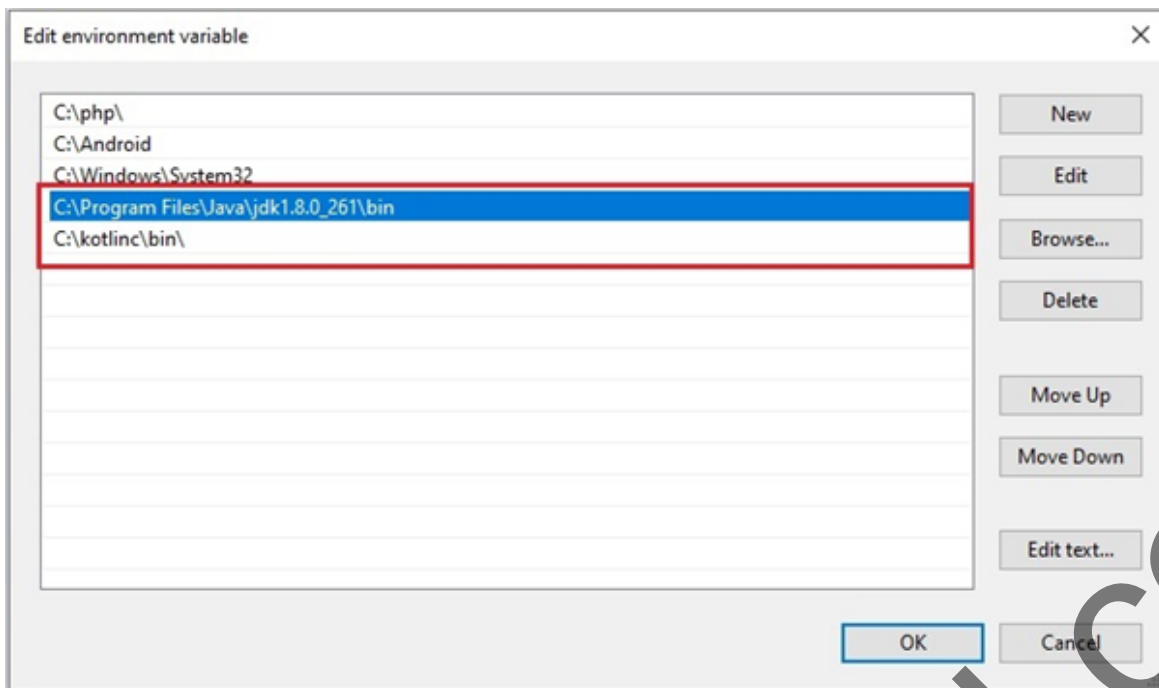
- Then, click on the **Environment Variables...** button.



- In the Environment Variables window, under **System Variables** select the Path variable and edit the variables using the **Edit...** button.



- Click on the **New** button and add the path of the bin folder of installed JDK and Kotlin folders.



To verify the installation, open command prompt and type java or javac command, if your installation is successful, you can see the output as shown below:

```
C:\WINDOWS\system32\cmd.exe
Microsoft Windows [Version 10.0.19041.928]
(c) Microsoft Corporation. All rights reserved.

C:\Users\TP>
Aug 09, 2021 4:47:46 PM org.jline.utils.Log logr
WARNING: Unable to create a system terminal, creating a dumb terminal (enable debug
logging for more information)
Welcome to Kotlin version 1.5.21 (JRE 1.8.0_261-b12)
Type :help for help, :quit for quit
>>>
```

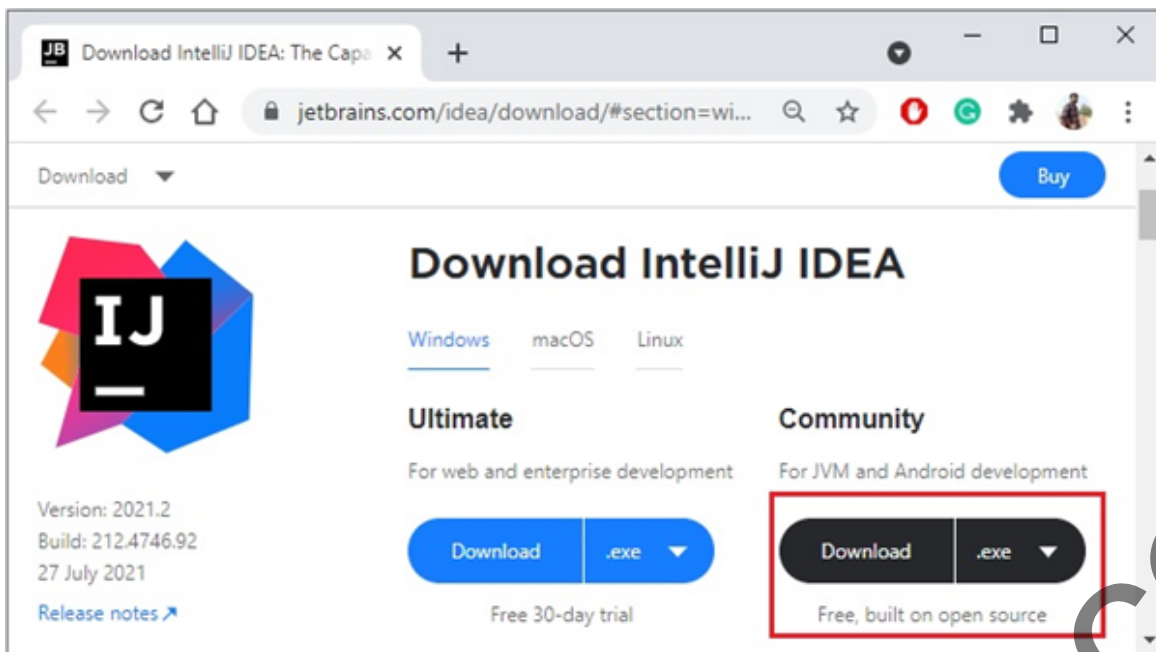
Setting Kotlin environment using IntelliJ IDEA

Kotlin is developed by the JetBrains that develops IDEs like AppCode, CLion, DataGrip, DataSpell, GoLand, IntelliJ IDEA etc.

The IntelliJ IDEA internally have Kotlin plugin bundled with it. To develop Kotlin download and install IntelliJ.

To install a recent version of IntelliJ IDEA:

- Open JetBrains Downloads page, you can download the free community edition.



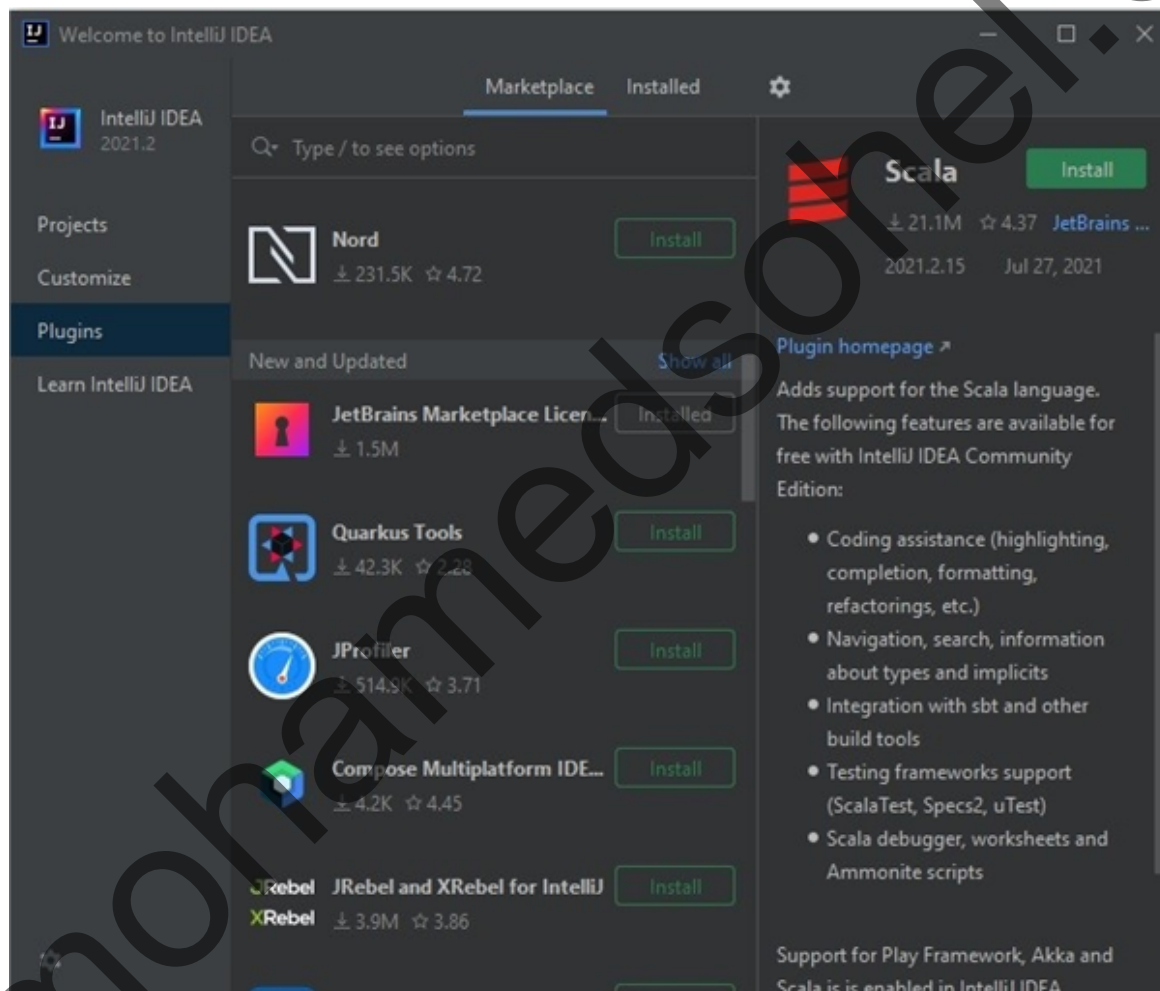
- If you run the downloaded file, it starts the installation process.



- Proceed with the installation by providing the required details and finally complete the installation.

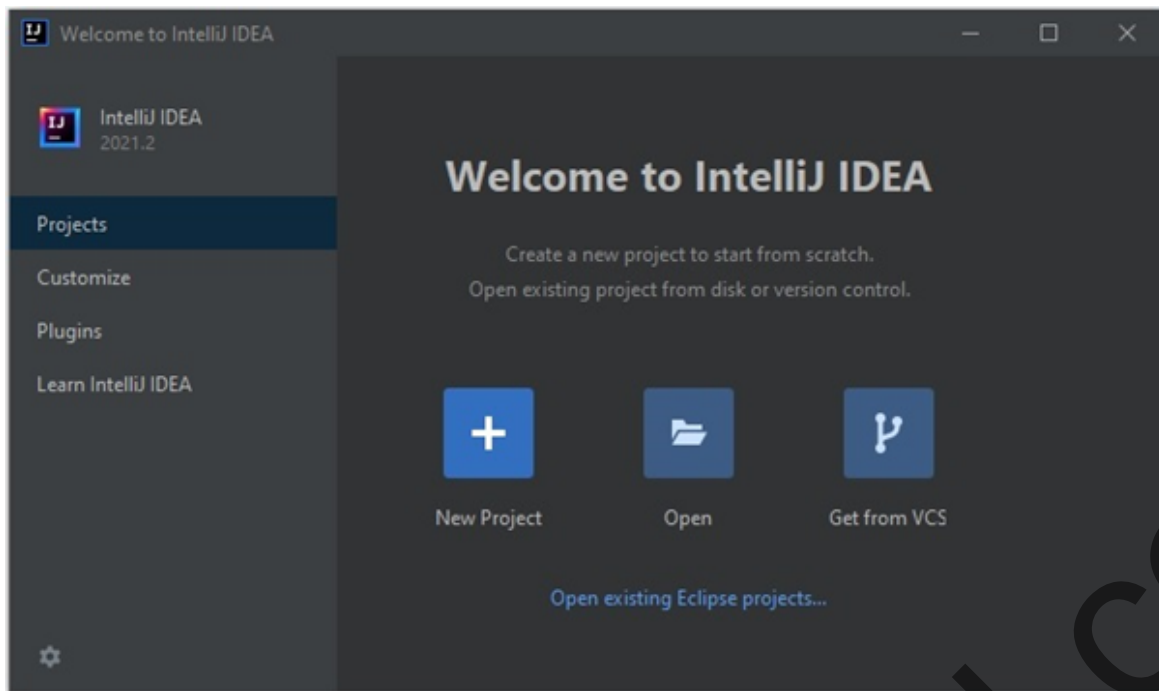


- The Plugins tab of IntelliJ displays all the available plugins. By default, Kotlin plugin is activated, in any case if it is not activated. Open the plugin tab, search for Kotlin and install it.

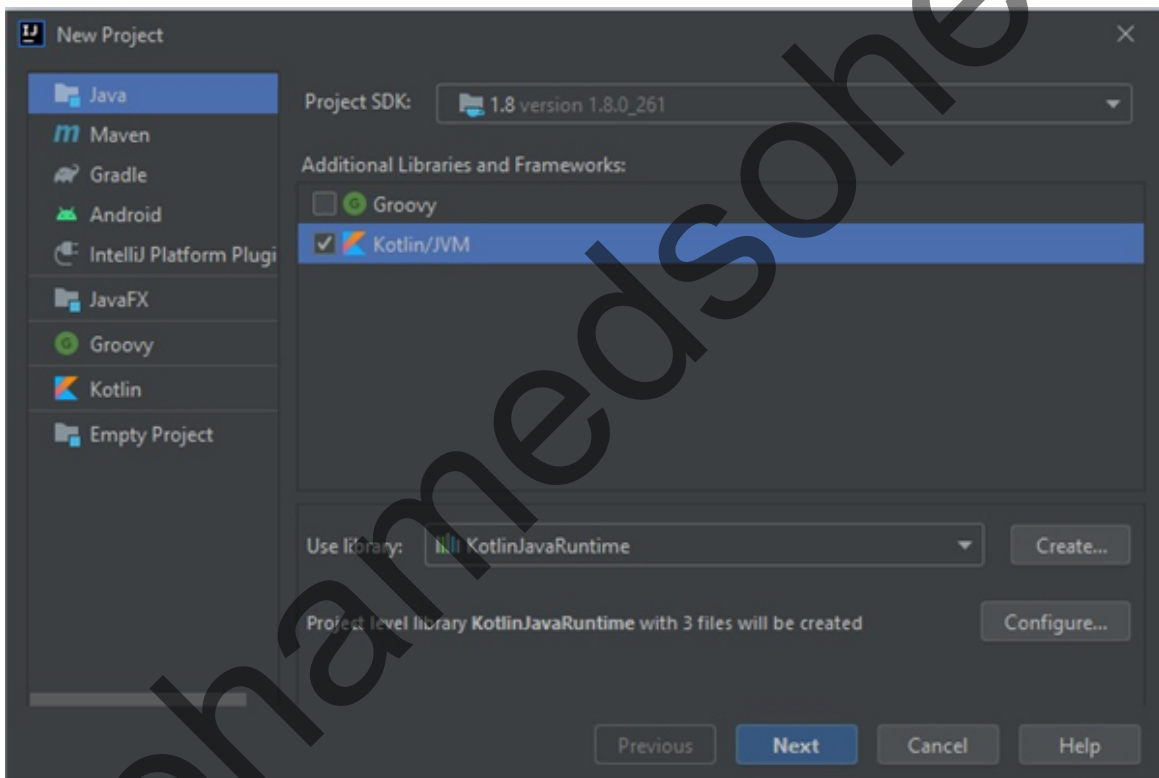


Creating first application

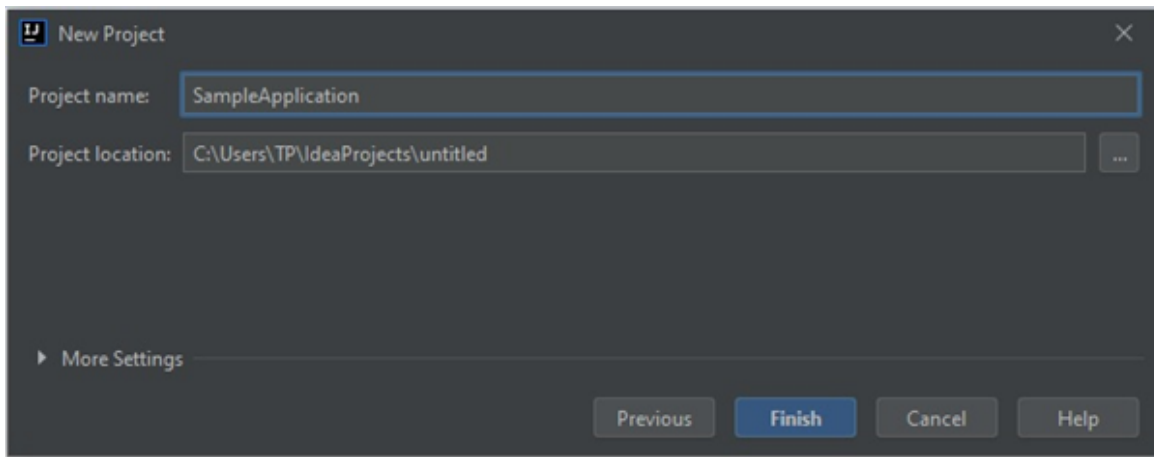
- To create first application, click on **NewProject**.



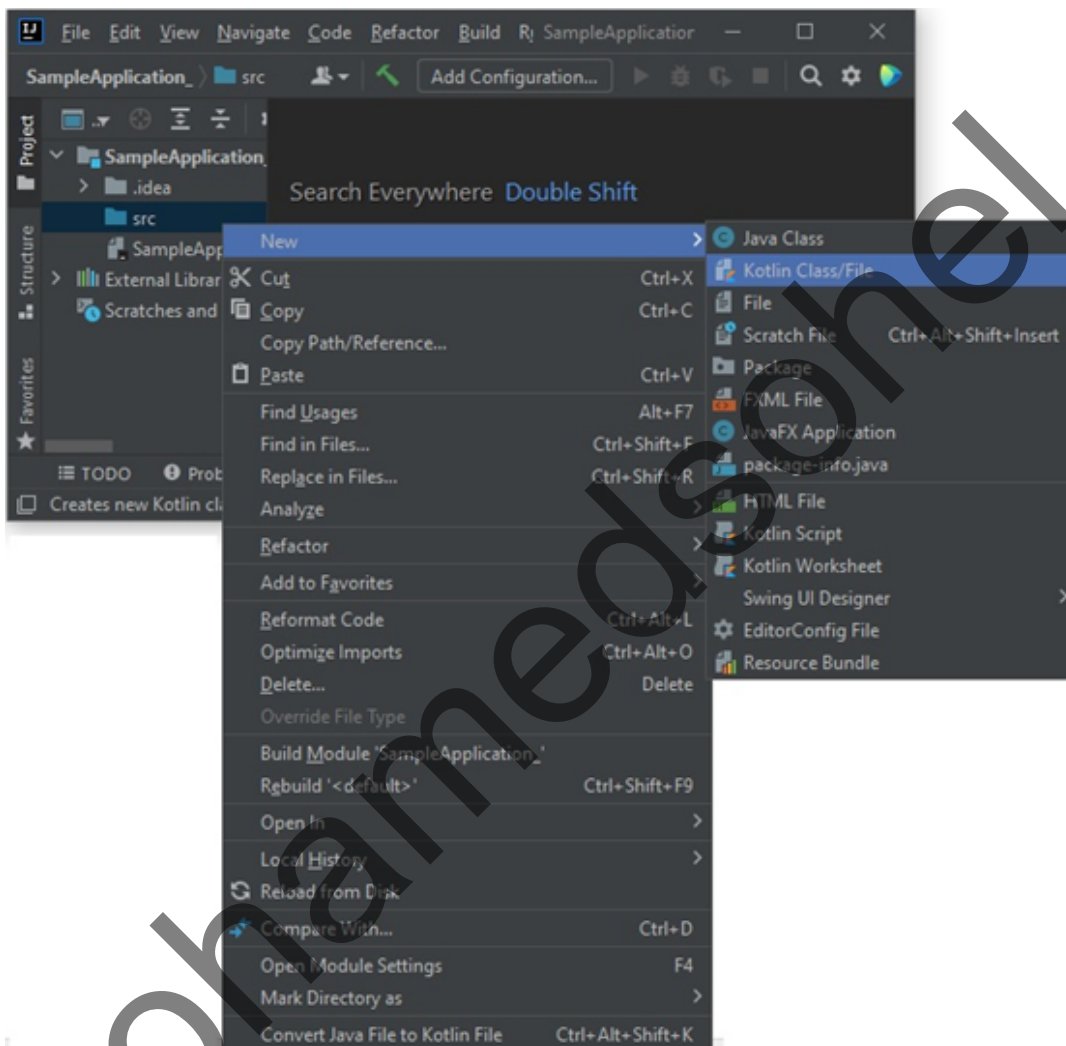
- Select **Kotlin/JVM** and click **Next**.



- Name the project and select the desired location.



- Now, create a new Kotlin file under the source(src) folder and let's name it as **Test**.



- You can create a sample function as shown below. You can run this by pressing **Ctrl + Shift + F10**.

